챕터 11, 12 과제 2019202050 이강현

선언된 1차원 배열의 접근

#include <stdio.h>

int main(void)

{

int arr[5]; //길이가 5인 정수형배열 선언

int sum = 0, i;//정수형 변수 sum,i 선언

arr[0] = 10, arr[1] = 20, arr[2] = 30, arr[3] = 40, arr[4] = 50;//배열 초기화

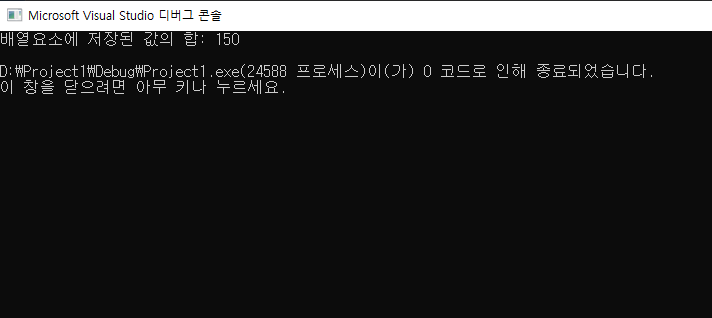
for (i = 0; i < 5; i++)//정수형 변수 i를 이용한 반복문

sum += arr[i];//모든 각각 배열들의 합

printf("배열요소에 저장된 값의 합: %d \n", sum);//총합을 출력

return 0;

}



한줄 토의: 배열을 선언할 수 있고 각 배열마다의 값에 관여할 수 있으며 모든 합을 반복문을 통해 구할 수 있다.

1차원 배열의 선언, 초기화 및 접근 관련 예제

#include <stdio.h>

int main(void)

{

int arr1[5] = { 1, 2, 3, 4, 5 };//길이가5인 정수형 배열 선언 및 초기화

int arr2[] = { 1, 2, 3, 4, 5, 6, 7 };//길이가 7인 정수형 배열 선언 및 초기화

int arr3[5] = { 1, 2 };// 길이가 5인 정수형 배열 선언 및 초기화

int ar1Len, ar2Len, ar3Len, i;//정수형 변수 4개 선언

printf("배열 arr1의 크기: %d \n", sizeof(arr1));//변수 arr1 바이트 크기 출력

printf("배열 arr2의 크기: %d \n", sizeof(arr2));//변수 arr2 바이트 크기 출력

printf("배열 arr3의 키기: %d \n", sizeof(arr3));//변수 arr3 바이트 크기 출력

ar1Len = sizeof(arr1) / sizeof(int); //배열 arr1의 길이 계산

ar2Len = sizeof(arr2) / sizeof(int); //배열 arr2의 길이 계산

ar3Len = sizeof(arr3) / sizeof(int); //배열 arr3의 길이 계산

for (i = 0; i < ar1Len; i++) //arr1 배열값 출력하는 반복문

printf("%d ", arr1[i]);

printf("\n");

for (i = 0; i < ar2Len; i++) //arr2 배열값 출력하는 반복문

printf("%d ", arr2[i]);

printf("\n");

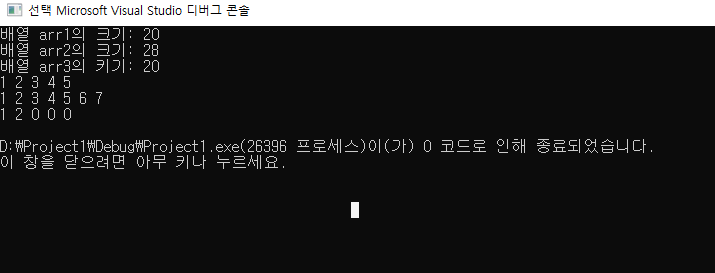
for (i = 0; i < ar3Len; i++) //arr3 배열값 출력하는 반복문

printf("%d ", arr3[i]);

printf("\n");

return 0;

}



한줄 토의: 배열의 크기를 정수형 변수을 이용해 표현할 수 있고 반복문을 통해 배열의 길이를 유추할 수 있다.

char형 배열의 문자열 저장과 널 문자

#include <stdio.h>

int main(void)

{

char str[] = "Good morning!"; //char형 str배열 선언 및 초기화

printf("배열 str의 크기: %d \n", sizeof(str));//str바이트 크기 출력문

printf("널 문자 문자형 출력: %c \n", str[13]);//str배열 14번째 문자 출력

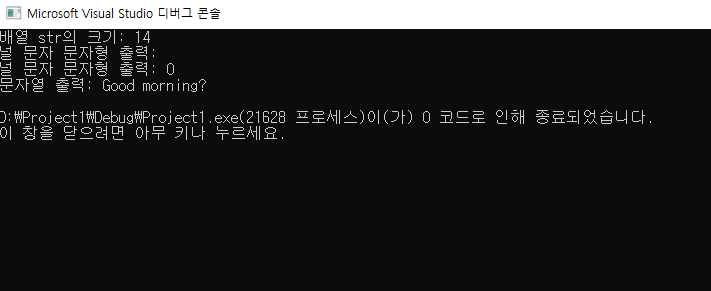
printf("널 문자 정수형 출력: %d \n", str[13]);//str배열 14번째 정수 출력

str[12] = '?'; //str배열 13번째값에 ?대입

printf("문자열 출력: %s \n", str);//str배열 전체 출력

return 0;

}



한줄 토의: 널 문자의 의미를 알 수 있다.

널 문자와 공백 문자의 비교

#include <stdio.h>

int main(void)

{

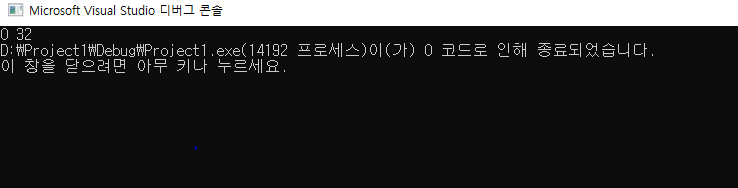
char nu = '\0'; //널 문자 저장

char sp = ' '; //공백 문자 저장

printf("%d %d", nu, sp); // 0과 32 출력

return 0;

}



한줄 토의: 널 문자와 공백문자의 차이점을 안다.

scanf 함수를 이용한 문자열의 입력

#include <stdio.h>

int main(void)

{

char str[50];//char형 길이가 50인 str 배열 선언

int idx = 0;//정수형 변수 idx를 선언후 0으로 초기화

printf("문자열 입력: ");//출력문

scanf("%s", str); //문자열을 입력 받아서 배열 str에 저장!

printf("입력 받은 문자열: %s \n", str);//출력문

printf("문자 단위 출력: ");//출력문

while (str[idx] != '\0')//scanf 함수의 호출을 통해서 입력 받은문자열의 끝에도 널 문자가 존재함을 확인하기 위한 반복문

{

printf("%c", str[idx]);

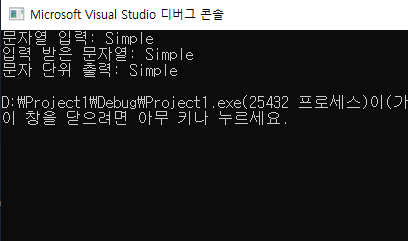
idx++;

}

printf("\n");

return 0;

}



한줄 토의: 문자열의 끝에는 널 문자가 존재함을 안다.

문자열의 끝에 널 문자가 필요한 이유

#include <stdio.h>

int main(void)

{

char str[50] = "I like C programing";

printf("string: %s\n", str);

str[8] = '\0'; //9번째 요소에 널 문자 저장

printf("string: %s \n", str);

str[6] = '\0'; //7번째 요소에 널 문자 저장

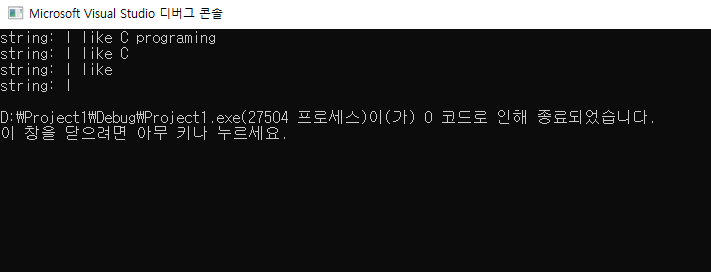
printf("string: %s \n", str);

str[1] = '\0'; //2번째 요소에 널 문자 저장

printf("string: %s \n", str);

return 0;

}



한줄 토의: 널 문자의 존재가 문자열의 끝을 의미한다는 것을 안다.

챕터 12

실습12-1

#include <stdio.h>

int main(void)

{

int num1 = 100, num2 = 100;//정수형 변수 num1,num2를 선언하고 100으로 초기화

int \*pnum;//포인터 변수 pnum선언

pnum = &num1;//포인터 pnum이 num1을 가리킴

(\*pnum) += 30;//num1+=30;과 동일

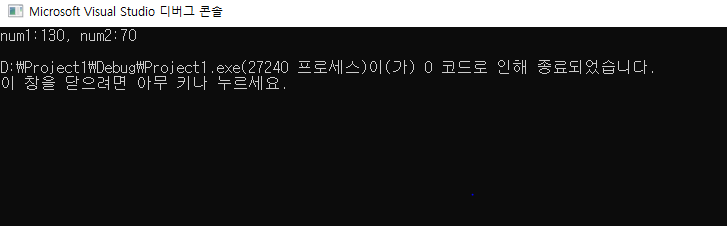
pnum = &num2;//포인터 pnum이 num2을 가리킴

(\*pnum) -= 30;//num1-=30;과 동일

printf("num1:%d, num2:%d \n", num1, num2);//num1,num2 값 출력

return 0;

}



한줄 토의: 포인터를 이용한 변수 값 변경이 가능함을 안다.

실습 12-2

#include <stdio.h>

int main(void)

{

int num = 10;//정수형 변수 num선언 및 10으로 초기화

int \*ptr1 = &num;//num을 가리키는 포인터ptr1선언

int \*ptr2 = ptr1;//ptr1을 가리키는 포인터 ptr2선언

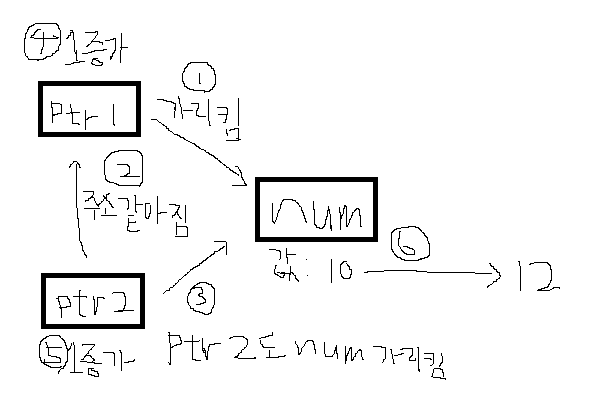
(\*ptr1)++;//num+=1과 동일

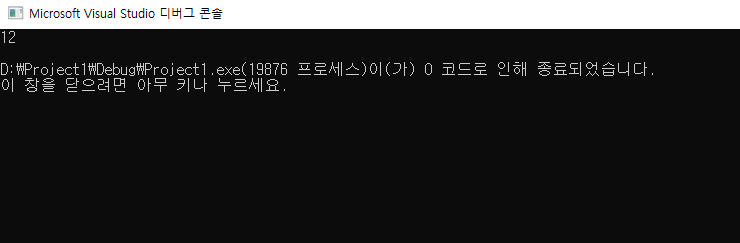
(\*ptr2)++;//num+=1과 동일

printf("%d \n", num);

return 0;

}





한줄 토의: 포인터 변수가 포인터 변수를 가리킬 수 있다는 것을 안다.

실습 12-3

//file: dereference.c

#include <stdio.h>

int main(void)

{

int data = 100;//정수형 변수 data선언 및 100으로 초기화

char ch = 'A';//char형 변수ch선언 및 A로 초기화

int \*ptrint = &data;//변수 data를 가리키는 포인터 ptrint선언

char \*ptrchar = &ch;//변수 ch를 가리키는 포인터 ptrchar선언

printf("간접참조 출력: %d %c \n", &ptrint, &ptrchar);//각각의 포인터가 가리키는 변수의 값 출력(간접 출력)

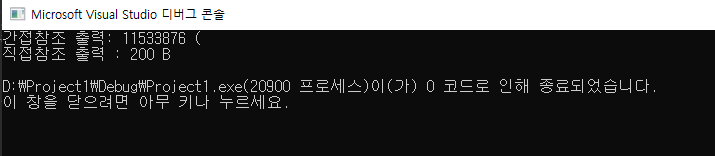
\*ptrint = 200;//변수 data를 \*ptrint로 간접참조하여 그 내용을 수정

\*ptrchar = 'B';//변수 ch를 \*ptrchar로 간접참조하여 그 내용을 수정

printf("직접참조 출력 : %d %c\n", data, ch);//변수의 값 직접 출력

return 0;

}



한줄 토의: 포인터 변수를 이용하여 변수의 간접참조 출력이 가능함을 안다.

실습 12-4

//file: nullpointer.c

#include <stdio.h>

int main(void)

{

int \*ptr1, \*ptr2, data = 10;//포인터 ptr1,ptr2선언과 정수형 변수 data선언 및 10으로 초기화

ptr1 = NULL;

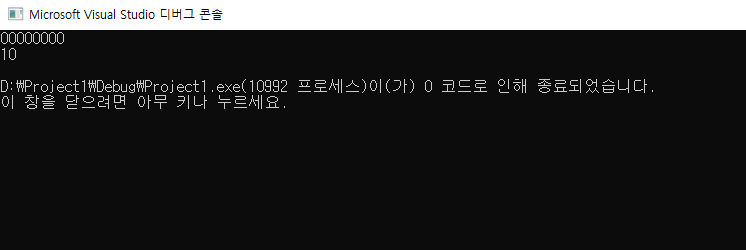
printf("%p\n", ptr1);//null을 대입하여 주소값이 0이 됨

//printf("%p\n",ptr2);

printf("%d\n", data);//출력문

return 0;

}



한줄 토의: 포인터에 null을 대입하면 주소값이 0이됨을 안다.

실습 12-5

//file: basicpointer.c

#include<stdio.h>

int main(void)

{

char c = '@';//char형 변수 c 선언 및 @로 초기화

char \*pc = &c;//c를 가리키는 char형 포인터 변수pc 선언

int m = 100;//정수형 변수 m선언 및 100으로 초기화

int \*pm = &m;//m을 가리키는 정수형 포인터 변수 pm 선언

double x = 5.83;//double형 변수 x선언 및 5.83으로 초기화

double \*px = &x;//x를 가리키는 doubel형 변수 px 선언

printf("변수명 주소값 저장값 \n");// 출력문

printf("----------------------------------\n");//출력문

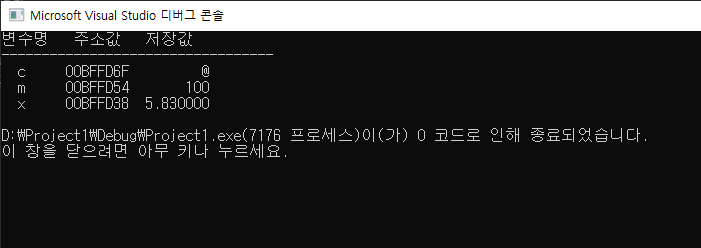
printf("%3s %12p %9c \n", "c", pc, c);//변수명, 주소값, 저장값 줄 맞춰 출력

printf("%3s %12p %9d \n", "m", pm, m);// 변수명, 주소값, 저장값 줄 맞춰 출력

printf("%3s %12p %9f\n", "x", px, x);// 변수명, 주소값, 저장값 줄 맞춰 출력

return 0;

}



한줄 토의: 포인터의 형은 포인터가 가리키는 변수의 형과 같아야함을 안다.

실습 12-6

#include <stdio.h>

int main(void)

{

int num1 = 10;//정수형 변수 num1 선언 및 10으로 초기화

int num2 = 20;//정수형 변수 num2 선언 및 20으로 초기화

int pnum;//ptr1과 ptr2가 가리키는 변수를 바꾸기 위한 변수

int \*ptr1 = &num1;//num1을 가리키는 포인터 ptr1선언

int \*ptr2 = &num2;//num2을 가리키는 포인터 ptr2선언

(\*ptr1) += 10;//ptr1이 가리키는 변수 10 증가

(\*ptr2) -= 10;//ptr2이 가리키는 변수 10 감소

pnum = \*ptr1;//ptr1과 ptr2가 가리키는 변수 서로 바꾸기

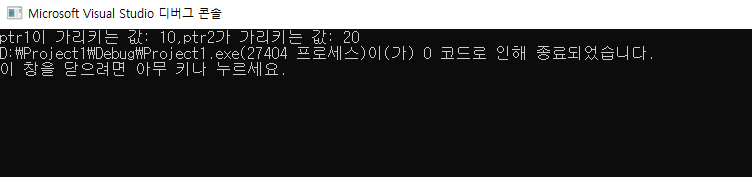
\*ptr1 = \*ptr2;

\*ptr2 = pnum;

printf("ptr1이 가리키는 값: %d,ptr2가 가리키는 값: %d", \*ptr1,\*ptr2);//출력문

return 0;

}



한줄 토의: 변수 선언과 포인터가 변수를 가리키게 하도록 코드를 짜고 포인터를 이용한 변수값 변경과 포인터들이 가리키는 변수들의 변경이 가능하다는 것을 안다.